A CONTROL SCHEME FOR MEASURE OF PERFORMANCE AND EFFICIENCY OF TACTICAL COOPERATIVE ROBOTS

Amir Shirkhodaie
Intelligent Tactical Mobility Robotic Research Laboratory
Department of Mechanical & Manufacturing Engineering
Tennessee State University
Ashirkhodaie@TNState.edu

ABSTRACT

We have discussed technical issues regarding supervised world perception modeling and task planning of cooperative mobile robots performing tactical tasks in unstructured environment. We have introduced a hierarchy Supervisory Controller for robust cooperative task deployment of heterogeneous semi-autonomous robotic vehicles. Primarily, we have described functional and modular architecture of the Supervisory Controller and presented our strategies for separation of supervisory functions according to their level abstraction, complexity, precedence, and intelligence. Furthermore, we have discussed control schemes for measure of performance and measure of efficiency of our cooperative robots tested under different task situations. Results indicate the hybrid Supervisory controller performs satisfactorily in most simulated cases.

Keywords: Cooperative Robots, Supervisory Control, Measure of Performance, and Measure of Efficiency.

1. INTRODUCTION

Cooperative robots are critically important for solving a number of real-world problems. Their applications span from civilian search and rescue missions to military battlefield reconnaissance and surveillance covert operations to deep space scientific research explorations. In very near future, potential commercial applications of cooperative robots are foreseen to be ubiquitous. The cooperative robots while benefit from inherent parallelism, their robust control is both multi-folded and challenging to achieve and sustain reliably. Many different techniques and approaches for mobile robots control have been developed. The proposed control techniques in literature can be, categorically, classified as either deliberative, reactive, or hybrid in nature.

The control schemes that tend to be more deliberative require relative more knowledge about the world. They use this knowledge to predict the outcome of their actions, an ability that enables them to optimize their performance relative to their model of the world. Deliberative reasoning about task planning of cooperative robots requires strong assumptions about this world model. Primary knowledge upon which the reasoning is based on, should be consistent, reliable, and certain. In a dynamic world, where objects may have arbitrarily moves (i.e., in a battlefield or a crowded hallway), it is potentially risky to rely on information that no longer be valid. Instead, world representational models are generally constructed at run-time using a combination of past knowledge gained about the environment and exteroceptive sensory data.

At the other end of the spectrum, the reactive control systems attempt to tightly couple perception and action in order to achieve faster robot response in dynamic and unstructured worlds while minimizing computational overhead. With a purely reactive control system, it is rather difficult to achieve planned deliberated tasks consistently. This is mainly due to variability in world uncertainty and lack of robot's knowledge and ability in resolving conflicts between competitive world perceptions in a given situation. In most cases, a wrongly selected world perception may cause the robot an unrecoverable deadlock situation or failure.

The temporal inconsistency and stability of the environment and the robot's immediate sensing inadequacy for a task are typically coupled. Difficulty in proper localization of a robot and the way that the robot perceives its surrounding world also yield possible situations that typically grounds erratic conflicts in decision making process of the robot. To demonstration this notion, consider figure 1 that illustrates spatial configuration of three cooperative robots. Should each robot presume the other two robots as obstacles or as its teammates approaching it unintentionally? One answer is it would be dependent on nature of the task and how the navigational modes of the robots are defined. If the robots had reactive behavior, they would probably try to find a way out of the crowd or avoid the deadlock situation. If the robots had deliberative behavior, each robot might refer to its chronicle memory and try to reason why the other robots are there in the first place before it makes any decision. Situation like this example can occur frequently during task execution of cooperative robots within a limited work environment. Hence, it is responsibility of an intelligent supervisory system to deal with such situations in a manner that causes least perplexity to plan execution of robots.

An overview of common conceptions of the behavior-based approaches is given by Mataric [1]. Brooks [2] describes four key concepts that lead to behavior-based robotic: situatedness, embodiment, intelligence, and emergence. The design of behavior-based systems is often referred to as a "bottom up" process, but this offers not so much to determination of the structure of the system as to a basis in physical sensing and

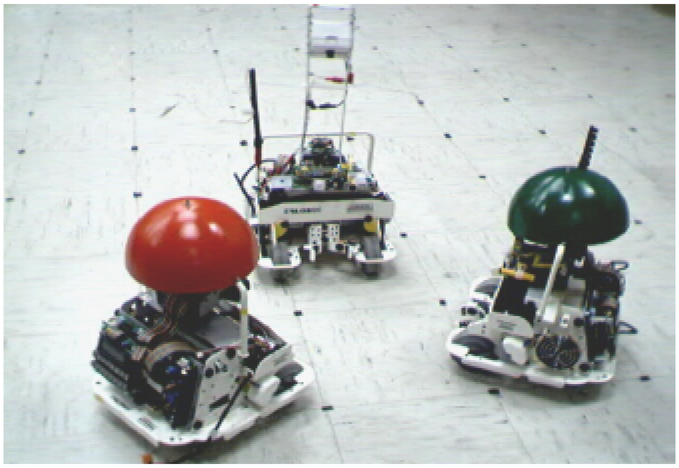

Figure 1. Three Semi-autonomous Cooperative Robots in a Tactical Formation.

action, and incremental development of sophistication from simple to complex. Namely, there are structured in terms of observable activity that they produce, rather than traditional functional decompositions. The activity producing components, behaviors, compete for actuator resources as well as share perceptions of the world rather than any centralized representation.

Recently, hybrid deliberative/reactive robotic architectures have emerged that rely more heavily on explicit world representations and tend to combine many aspects of traditional AI symbolic methods with situated-based reasoning. They operate based on abstract representation of the world in light of providing faster response, better robustness, and more tractable than deliberative and reactive systems. Hybrid architectures permit reconfiguration of reactive control systems based on available world knowledge through their ability to reason over the underlying behavioral components. However, building of such hierarchy systems requires compromise and full utilization of reactive and deliberative systems to maintain the desirable system performance and efficiency.

An overview of approaches and issues in cooperative robotics has been also reported in [3,4,5]. Parker [6] has demonstrated multi-robot target observation using the ALLIANCE Architecture, where action selection consists of inhibition (through motivation behaviors). As opposed to her architecture, Pirajanian and Mataric [7] have developed an approach to multi-robot coordination in the context of cooperative target acquisition. Their approach is based on multiple objective behavior coordination extended to multiple cooperative robots. They have demonstrated a mechanism for distributed command fusion across a group of robots to pursue multiple goals in parallel. This technique enables individual robot to select actions that no only benefit itself but also benefit the group as a whole. A significant amount of work in this area is being conducted by researchers at NIST. Their hierarchical control architecture is shown to have capability in controlling several unmanned mobile robotic vehicles in unstructured environment using a hierarchy platoon level control scheme where the robots follow a designated leader while maintaining a fair distance apart [14].

Nonetheless, intelligent control of multi-agent robots is both complex and challenging. The complexity of the task is contributed to a number of compounding factors including: multi-agent task decomposition, task distribution, resource allocation, sensory world perception modeling and data sharing, pattern recognition and reasoning, skill learning and adaptation, communication networking, man-machine interaction, and others. For intelligent strategic task planning, execution, and monitoring of cooperative robots, one should be concerned with many of above technical challenges.

In this paper, we will present a hybrid hierarchical deliberative/reactive robotic architecture called, *Supervisor Mobility Controller* - in short *"Supervisor"* for controlling a team of cooperative robots. By combining reactive and deliberative navigational schemes, we have created a set of group navigational techniques assisting task deployment of the robots. The Supervisor has been tested for localization and dynamic cooperative task planning of robots. Performance and efficiency of the system is measured on a physical robotic system consisting of five cooperative robots.

The proposed Supervisor control system has been developed under FMCell software [18]. FMCell provides tools for world perception construction and sensors modeling in 3D virtual simulation environment. Other features of the software include: high-level object-oriented environment with embedded robot behavioral modeling tools, fast image processing tools, AI-based inference engines for knowledge processing and reasoning, and soft computing developmental tools such as neural networks, fuzzy logics, and genetic algorithms for modeling, simulation and validation of control strategies.

2. SUPERVISOR ARCHITECTURE

The Supervisor was originally designed for control of semi-autonomous robots operating under one central control unit. The modular software implementation of the Supervisor is presented in Figure 2. Supervisor has a hierarchical architecture and designed to handle hybrid reactive/deliberative task deployment of the cooperative robots. Detailed description of different functions of this Supervisor is behind the scope of this paper and can be found elsewhere [8,9]. The hierarchy architecture consists of sensing, planning and acting components. The system allows direct interaction of the human operator at different levels of abstract task planning, execution, and monitoring with minimum restriction. An exclusive language allows for mission plans of the virtual robots with concise details. The supervisor can be used for control of both simulated and physical cooperative robots .. The same syntax as used for programming of physical robots is used for programming of the simulated robots. This feature significantly reduces development and implementation times and makes it submissive to both software and hardware requirements.

There are many good reasons for working with simulated robots: “learning often requires experimenting with behaviors that may occasionally produce unacceptable results,” [10]. Making mistakes on real physical robots can be sometimes quite costly and dangerous. Nolfi et. Al. [17] suggests that simulated robots be used in developing control systems for real robots when certain special conditions are taken into account. However, one should not expect control systems that are evolved in a simulated environment to behave exactly the same in a real environment. Environment has many degrees of uncertainties, in particular in case of mobile robots where skidding and slippage are inevitable. To reduce this ambiguous problem, one can literally add certain degree of white noise to parameters having anticipated uncertainties. For instance, in dead-reckoning computation of robots, one may add random noise to offset encoders and compass readings of the robots and mimic irregularity in mobility behavior of the simulated robots modeled as if they are operating in unstructured environment.

In our approach, we consider simple unknown world with geometrically identifiable obstacles. Construction of the world perception is achieved at three levels. Initially, fragmented sensory data gathered by robots are filtered, analyzed, and streamlined into a more geometrical representation, i.e., lines, and curves. Next, robots are localized through using a localization technique [15]. Figure 3 illustrates gradual world perception construction of a robot navigating a simulated indoor environment. Secondly, localization of robots within the environment is performed to localize the robot via a target tracking technique [16]. The refined world perception information along with estimated global positions of the located robots are registered in a lookup table for further processing. From the world perception information, we extract world geometrical features. A connectivity check is performed to

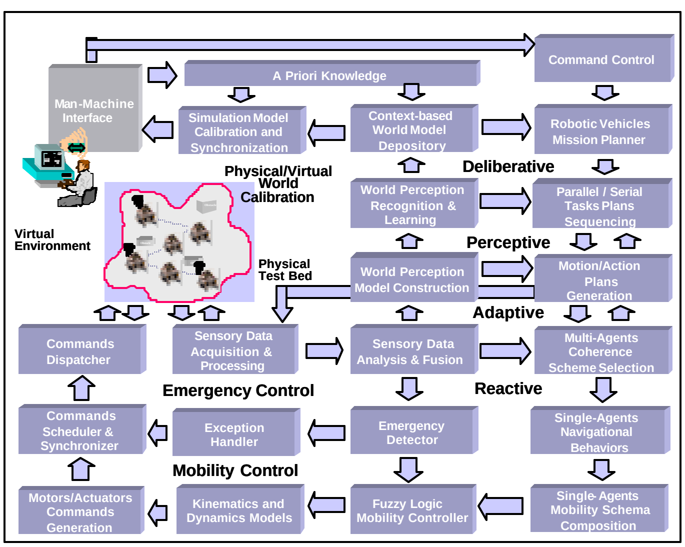

Figure 2. Modular Architecture of Supervisor Mobility Controller

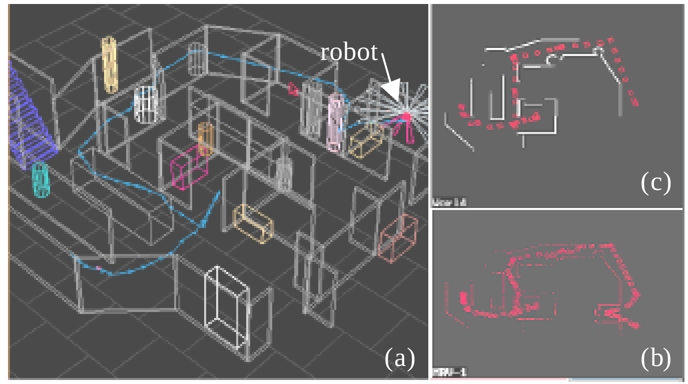

Figure 3. (a) A Simulated Robot World, (b) Reflection of Robot’s Range Sensors Data as the Robot Navigates its Path, (c) Constructed Robot World Model Based on Sensory Data

identify neighboring line segments that closely represent a certain environment feature. The line segments belonging to a detected environment feature are marked accordingly along with a degree of certainty in another lookup table. To infer a robot's task plan and the world model, we use the latter lookup [16].

For deliberative task planning, cooperative robot team members must be planned based on a certain governing behavioral scheme. Some approaches in literature considered evolution of such governing behaviors as a way of demonstrating the cooperation among robots in an emergence way [10]. Some others attempted to motivate robots into cooperation [6]. While others have proposed consenting agents with mechanism for negotiation among cooperative agents [11]. For practical purposes, algorithmic techniques have shown good stability and performance [12,13]. We have developed different cooperative robots behavior-based algorithms for different applications [19-21].

Supervisor supports different multi-agent robot platform, (see Figure 4) and a number of cooperative navigational deployment strategies. Each robot by itself can be assigned variety of distinct behavior-based navigational schemes while engaging in a cooperative task. Each subsumption-based navigational behavior controls interaction of a robot with the environment and arrives at common decisions regarding turning and steering requirements of the robot for a given situation.

Supervisor handles task planning of cooperative robots using a high-level object-oriented language that will be discussed shortly. Supervisor recognizes two groups of task instructions. The first group of task instructions deals with mobility requirements of robotic vehicles and the other group deals with sensory requirements of cooperative robots (i.e., an instruction that activate the robot to perform an environment sensing operation).

Supervisor handles the Deliberative tasks with the highest priority while paying coarse attention to detailed of mobility of robots. In coordinated task planning of cooperative robots, coherence task planning of cooperative robots is very critical.

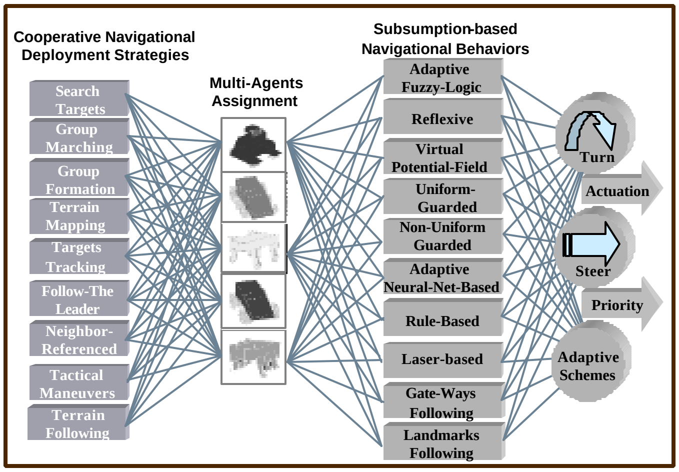

Figure 4. Cooperative Behaviors and Behavior-based Navigational Alternative of Multi-Agent Robots Supported by the Supervisor Mobility Controller.

Equally important, is the interaction between deliberative and reactive behavioral arbitration of cooperative robots. At the highest level of task planning, the mission plan can be designed using structured syntax and semaphore directives. Refer to Figure 5, for an example of cooperative robot task programming scheme. Supervisor has a parser that interprets abstract textual task commands as shown in the example. A double linked list buffers all task instruction with a time stamp.

```
Def  Task1
Channel: 1,2,3                         // Assign Channel 1,2,3 For Comm
Vel: 5,5,5                     // Assign Velocity 5 in/s to Robots 1,2,& 3
Acc: 3,3,3                   // Assign Accelecration in/s2 to Robots 1,2,3
Loc:  (10,30),(40,30),(60,-20)                  // Assign 3 Vectors for the
                                                  // the robots to follow.
NavBeh:  1,2,1                 // Set Navigation Behaviors of Robot 1, 2,
                             //& 3 to Navigational Id # 1,2, 1 Respectively.
MC: 1                                  //Turn on Continuous Processing Mode
Go: 1,1,1                               // Execution queued motion commands
                                                // for Robots 1, 2, and 3.
Wait  t >4000                           // Wait until 4000 ms time is elapsed.
CapImg: 1,,1                              //Have Robots 1 and 3 capture images
                                                    // of their direction.
LMScan: ,1,                                 //Have Robot 2 Scan Using Laser
                                              //Measurement Range Finder.
Go: 1,1,1                   //Now, have all 3 robots to perform their sensing.
WaitAck: 1,1,1                            //Wait Until All Robots Acknowledge
                                          //Completion of Their Specific Tasks.
MC: 0                                  //Turn off Continuous Processing Mode
EndDef                                               //Terminate Task Block.
```

Figure 5. An Example of Task Programming of Cooperative Robots .

For synchronization purposes, all motion commands and sensory data acquisition operations are buffered in a temporary transit queue until a trigger statement is executed. Execution of buffered commands is performed in the order that the commands have been received. In the example below, for instance, some preparatory commands set up communication channel, and preset velocity and acceleration of robot. The command *Loc* assigns three positional vectors to the robots to follow. The parameters of the *Loc* function define position and heading angle requirement of corresponding robot in inch and degree respectively. Next, proper navigational behaviors are assigned to individual cooperative robots. Instruction *Go* causes execution all buffered task plans to begin with the predefined task requirements, i.e., robots should presume a velocity 5 in/sec and an acceleration of 3”/sec^2. With continuous processing mode on, execution of commands proceeds right after executing Go statement without any delay. Next, the task command processing is delayed for 4 seconds before robots 1 and 3 are assigned to capture image along their heading direction, while the robot 2 is assigned to scan its heading for detection of obstacle. The last Go statement causes transmission of proper sensory data acquisition instruction to robots.

3. CONTROL SCHEMES FOR TASK PLAN TESTING OF COOPERATIVE ROBOTS

In practice, a mission plan may comprise of many sub-task plans – requiring cooperative to do many tasks either in synchronization or independent but in harmony. Each sub-task plan may consist of many symbolic notions of activities. Figure 6 shows one such sub-task plan where four robots are employed to rapidly create a consolidated world perception of their unknown environment. In this scenario, each robot's behavior is set to be reactive. Without explicitly defining individual navigational task of each robot, we applied a simple deliberative learning strategy. In this technique robots are rewarded more for exploring unvisited area of the world. The Supervisor performs two operations in this scenario - gathering range data from individual robots and fusing the range data to

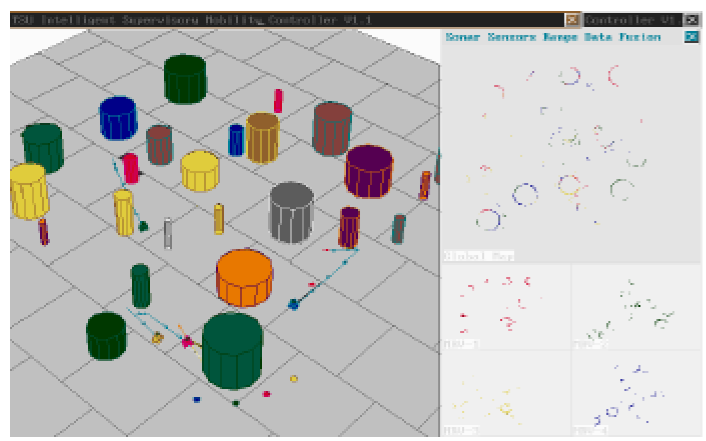

Figure 6. (a) A Simulated Scenario of Cooperative Robots World. (b) World Perception Model Based on Range Sensor Data Fusion of Cooperative Robots (c) World Perception of each of four individual Robots.

construct a world perception model as shown in the upper right hand corner of the figure 6. As the sonar range data become available, the world model is progressively constructed. The world model is partitioned into matrix of cells. Each cell will have a certain potential depending upon total number of sonar data point that it contains. The Supervisor creates a gradient field based on potential of each cell. The robots are then command to explore world environment with low gradient slope. To encourage the robots to explore the entire area, they are rewarded more for exploring the areas that they have not visited before. By adjusting the rewarding weights, the navigational behavioral of the robot are tuned. To prevent the robots from over exhaustion in their search, a time-based terminating condition is imposed that is if the new world discovery slows than behind a threshold over a fixed period of time, then the navigational search should stop. The algorithm was tested on a team of five cooperative robots. One robot from the group becomes as anchor and monitors operations of its other team members using its on-board surveillance camera. Localization of cooperative searching robots is performed using an image processing technique that localizes the cooperative robots in the image frame of the surveillance robot. The physical robotic test bed is shown in Figure 7. Coordinates of the robots in the image frame are next mapped to the world coordinate with the center of the camera at the origin [22]. A total of twenty tests were conducted to assess performance of the cooperative robotic team in detecting a total of 10 cans of cokes randomly located on the floor within an area of 20'x20'.

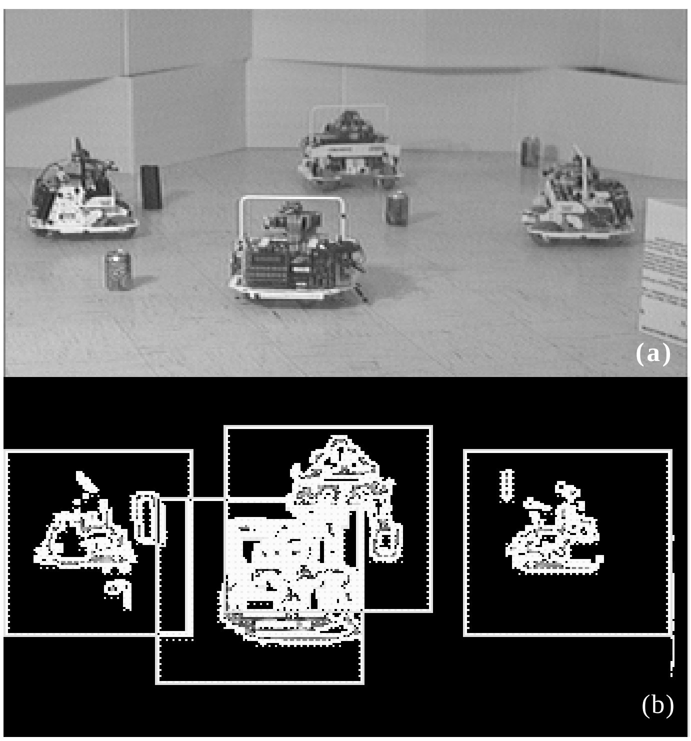

Figure 7. (a) Physical Cooperative Robot Test Bed. Localization of Cooperative Robot Using Visual Servoing Technique.

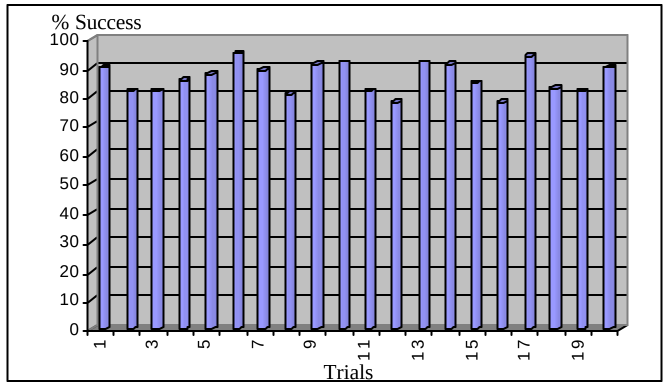

Figure 8. Performance Measure oCooperative Robots in Detecting Random Targets.

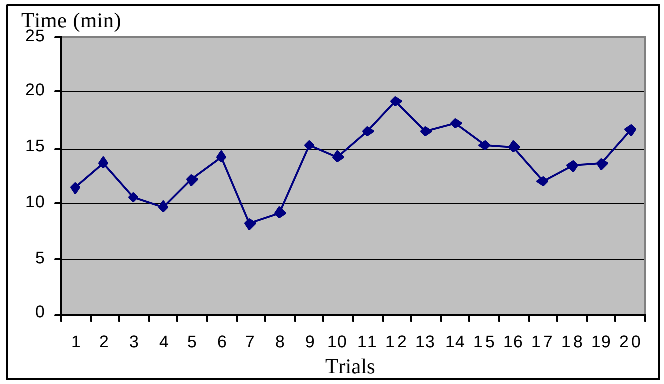

Figure 9. Time Efficiency of Cooperative Robots in Detecting Random Targets.

Figure 8 shows the results of experimentation. On average over 86.5 percent of time, all 10 targets were located with 99 percent confidence level. Figure 9 indicates the total time efficiency of the system in detecting all targets. While time efficiency of the system can serve as a basis for evaluation of intelligence of the robots, but some small adjustment in the control behaviors of the robots can have significant effect on overall efficiency of the system. Perhaps for reducing wondering time delay of the robots.

4. CONCLUSION

Cooperative robots have many practical applications. In this paper, we have discussed architecture of a Supervisory Mobility Controller with capability to facilitate deliberative/ reactive task deployment of cooperative robots. Some of the issues regarding robot's intelligence requirement at robot platform level and at cooperative robots level were discussed. To have fully functional cooperative robots, many research issues need to be addressed and taken into consideration. At present time, there is no single established standard or in testing procedures of the cooperative robots' intelligence. Measure of performance and efficiency of intelligent robotic system are very subjective and conditional. Minute adjustments in control parameters of a system can have significant influence over performance and efficiency of the system. Furthermore, incompatibility and heterogeneity among the robots makes it quite difficult to relate performance and efficiency of one system to another - even from one robot to another in the same platform and ranking.

5. ACKNOWLEDGEMENT

This work was partly funded by the US Army TACOM under DAAE07-01-C-L-065 and Penn State Applied Research Laboratory Under Prime Grant Number DAAD19-01-1-0504 from the US Army Research Office. The support from both funding agencies is acknowledged.

6. REFERENCES

[1] Mataric, M., "Behavior-based Control: Examples From Navigation, Learning, and Group Behavior," J. of Experimental and Theoretical AI, 9, 1997.

[2] Brooks, R., "Intelligence Without Reason," Computers and Thoughts - IJCAI 91, 1991.

[3] Arkin, R. C., Behavior-Based Robotics. The MIT Press, Cambridge, MA, 1998.

[4] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B., "Cooperative Mobile Robotics: Antenedents and Directions, Autonomous Robots, 4(1): 7-27, 1997.

[5] Mataric, M.J., "Issues and Approaches in the Design of Coolective Autonomous Agents. Robotics and Autonomous Systems," 16 (2-4), pp. 321-331, 1995.

[6] Parker, L. E., "Cooperative Robotics for Multi-target Observation," Intelligent Automation and Soft Computing, 5(1): 5-19, 1999.

[7] Pirjanian, P., and Mataric, M. J., "Multi-robot Target Acquisition Using Multiple Objective Behavior Coordination," in Proc. Of the IEEE International Conf. on Robotics & Automation, San Francisco, 2000.

[8] Shirkhodaie, A., Goetz R., Patkar, A., Mansour M. Aderogba, S., and Avalareddy N., "Behavior-Based Tactical Navigational And Mobility Control of Tandem Ground Robotic Vehicles", in the Proc. of SPIE Unmanned Ground Vehicle Tech. II, Orlando, FL, 2000.

[9] Shirkhodaie, A., Rangavajhala, S., "Supervised Tactical Mobility Behavior Modeling And Control of Multi-Agent Robotic Vehicles," in the Proceedings of SPIE, Unmanned Ground Vehicle Technology II, April 24-25, 2001, Orlando, FL.

[10] Liu, J., and Wu, J., Multi-Agent Robotic Systems, The CRS Press International Series on Computational Intelligence, 2001.

[11] Rosenschein, J. S., "Consenting Agents: Negotiation Mechanisms for Multi-Agent Systems," in Proc. of Int. Joint Conf. On AI, pp. 792-799, 1993.

[12] Miyata, N. Ota, J., and Aiyama Y., Asama, H.., and Arai, T., "Cooperative Transport in Unknown Environment - Application of Real-time Task Assignment," Proc. of the 2000 IEEE Int. Conf. On Robotics and Automation, pp. 3176-3182, 2000.

[13] Mansour, M., and Shirkhodaie, A., "Navigational Behaviors Fusion for Multi-Agent Intelligent Tandem Mobile Robots," in the 2000 Japan-USA Symposium on Flexible Automation International Conference on Manufacturing Systems" Innovation for the 21st Century, Ann Arbor, MI, July 23-26, 2000.

[14] Lacaze, A., "Hierarchical Architecture of Coordinating Ground Vehicles in Unstructured Environments," in Proc. of 2001 PERMIS Workshop on Measuring Performance and Intelligence of Systems, September 4, 2001.

[15] Hajjawi, M., and Shirkhodaie, A., "Comparative Visual Team Working and Target Tracking of Mobile Robots", in the 34th IEEE Southeastern Symposium on System Theory, March 17, 19, 2002.

[16] Rangavajhala, S., and Shirkhodaie, A., and "Visual-Servoing and Navigation of Mobile Robots For Target Tracking Applications With Laser Sensory Data Fusion," in the Proceedings of the IEEE SSST 33rd Southeastern Symposium on System Theory, March 18-20, 2001.

[17] Nolfi, S., Evolutionary Robotics: Exploiting the Full Power of Self-Organization, Connection Science, pp. 167-183, 1998.

[18] Shirkhodaie A., "FMCell: User Manual - Vol I and II," Tennessee State University, Department of Mechanical and Manufacturing Engineering, April 2002.

[19] Adergoba, S., and Shirkhodaie, A., "Navigation and Visual-Terrain Tracking of Multiple Mobile Robots With Vision and Sonar Using Neural Network Techniques," in the IMCC'2000 International Manufacturing Conference in China," August 16-17, 2000, China.

[20] Avalareddy, N., and Shirkhodaie, A., "Communication-based Tactical Task Planning of Cooperative Tandem Mobile Robot Vehicles Under an Intelligent Supervisory Mobility Controller," in the 2000 Japan-USA Symposium on Flexible Automation International Conference on Manufacturing Systems", Ann Arbor, MI, July 23-26, 2000.

[21] Avalareddy, N., Shirkhodaie, A., "Cooperative Task Planning, Execution, and Monitoring of Mobile Robots Under a Supervisory Mobility Controller", in the IEEE Southeastcon 2000 Conference, Nashville, TN, April 7-9, 2000.

[22] Hajjawi, M.,"Visual Servo Tracking and Localization of Cooperative Robots," M.S. Thesis, Mechanical Engineering, Tennessee State University, August 2002.